

PDF フォーム

1. PDF フォーム機能の概要	5
1-1. PDF フォーム機能の概要	5
1-2. Create! FormCollect が提供する主な機能	5
2. フィールドの作成	6
2-1. フィールドを作る	6
フィールドオブジェクトの定義方法	6
フィールドオブジェクト属性ダイアログの構成	6
2-2. フィールドの外観、表示状態を設定する	7
フォントを設定する	7
背景色を設定する	8
境界線を設定する	8
フィールドの表示、印刷時の状態を設定する	9
フィールドを読取専用にする	9
フィールドへの入力（選択）を必須にする	10
2-3. テキスト入力を行うフィールドを作る - テキストフィールド	10
初期値を指定する	10
テキストの配置を指定する	11
入力可能な最大文字数を指定する	11
複数行のテキスト入力を可能にする	11
フィールド内でスクロールを可能にする	11
パスワード入力を行うフィールドを作る	11
マス目に区切られたフィールドを作る	12
入力時にスペルチェックを行う	12
出力時にフィールドに出力するデータが空白の場合、初期値を出力する	12
2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド	13
項目を追加（編集・削除）する	13
外部の CSV ファイルから項目をインポートする	14
初期選択項目を指定する	15
登録した項目を並べ替える	15
項目を編集可能にする	15
項目のスペルチェックを行う	15
2-5. メニューリストを作る - リストボックスフィールド	16
複数の項目を選択可能にする	16
項目を追加（編集・削除）する	16
外部の CSV ファイルから項目をインポートする	16
初期選択項目を指定する	16
登録した項目を並べ替える	16
2-6. ラジオボタンを作る - ラジオボタンフィールド	17
チェックマークのスタイルを指定する	17
選択時の値を指定する	17
デフォルトで選択状態にする	18

2-7. チェックボックスを作る - チェックボックスフィールド	18
チェックマークのスタイルを指定する	18
チェック時の値を指定する	18
デフォルトで選択状態にする	18
2-8. ボタンを作る - ボタンフィールド	19
ボタンに表示するテキストを指定する	19
2-9. 送信ボタンを作る - サブミットボタンフィールド	20
ボタンに表示するテキストを指定する	20
フィールドデータを処理するサーバスクリプトの URL を指定する	20
送信時のデータ形式を指定する	20
データを送信する対象のフィールドを指定する	21
2-10. クリアボタンを作る - リセットフォームボタンフィールド	21
ボタンに表示するテキストを指定する	22
フィールドデータの内容をリセットする対象のフィールドを指定する	22
2-11. 電子署名を行う領域を作る - 署名フィールド	23
署名を行った後にフィールドを読取専用にする	23
署名を行った後に任意の動作を行わせる	24
3. フィールドに特殊機能をつける	25
3-1. 入力したテキストの表示形式を指定する - フォーマット	25
入力したテキストを数値表記する	25
入力したテキストをパーセント表記する	26
入力したテキストを日付表記する	27
カスタム JavaScript を使用して独自の表示形式を指定する	28
3-2. 他のフィールド間の値の計算結果を表示させる - 計算	28
フィールドを指定して計算に使用する	28
任意の値を計算に使用する	29
カスタム JavaScript を使用して独自の計算方法を指定する	30
3-3. 入力した（選択した）フィールドの値を検証する - 検証	30
フィールドの値の範囲を数値で指定する	31
カスタム JavaScript を使用して独自の検証方法を指定する	31
3-4. メニュー項目の選択の変更時に任意の処理を行う	32
カスタム JavaScript を使用して選択項目の変更時の動作を指定する	32
3-5. アクションを組み込む	32
フィールドにアクションを組み込む	32
ページにアクションを組み込む	33
アクションを組み込む	34
JavaScript を組み込む - JavaScript の実行	34
フィールドデータを送信する - フォームデータの送信	34
フィールドデータをリセットする - フォームデータのリセット	36
フィールドの表示状態を切り替える - フィールドの表示 / 非表示	36
指定した URL に移動する - WWW リンクを開く	37
印刷ダイアログを起動する - ビューアメニューの実行	37
フォームデータをインポートする - ビューアメニューの実行	38
4. 外部からのデータを反映させる	39
4-1. フィールドデータを外部データから指定する	39

テキストフィールドに表示するテキストを外部データから指定する	39
選択状態にするメニュー項目を外部データから指定する	39
チェックマークを入れる項目を外部データから指定する	39
フォームデータの送信先 URL を外部データから指定する	39
4-2. フィールドオブジェクトにデータマッピングを行う	40
マッピング方法	40
マッピングデータ	40
4-3. データマッピングを行わない場合に反映されるフィールドの値	40
5. Tips	42
5-1. 出力時にフィールドの外観、状態を切り替える - FormSwitch オプション	42
出力パターン別にフィールドの外観設定を行う	42
出力パターン別に PDF フォームのプレビューを行う	43
出力パターン別に PDF ファイルの出力を行う	44
5-2. メニュー項目を外部データから取り込む	44
5-3. 指定したサーバスクリプトにフィールドデータを送信する	45
送信されるフィールドデータ	45
データ送信形式とその処理方法の例	46
5-4. ツールチップを表示させる	47
5-5. Tab キーで移動する順序を指定する	47
選択したフィールドのタブ順序を先頭に指定する	48
選択したフィールドのタブ順序を最後尾に指定する	48
選択したフィールドのタブ順序を1つ上げる	48
選択したフィールドのタブ順序を1つ下げる	48
選択したフィールドのタブ順序を任意の位置に指定する	48
5-6. 計算を行う順序を指定する	49
選択したフィールドの計算順序を先頭に指定する	50
選択したフィールドの計算順序を最後尾に指定する	50
選択したフィールドの計算順序を1つ上げる	50
選択したフィールドの計算順序を1つ下げる	50
選択したフィールドの計算順序を任意の位置に指定する	50
5-7. JavaScript を使用する	51
JavaScript をコーディングする	51
JavaScript をデバッグしたい	52
Acrobat JavaScript について	53
5-8. JavaScript 関数を登録する	54
JavaScript 関数を管理する	54
JavaScript 関数を編集する	55
5-9. JavaScript を一括して編集する	56
一括編集ダイアログを使用して JavaScript 関数を編集する	57
一括編集ダイアログを使用してカスタム JavaScript を編集する	57
5-10. JavaScript リファレンスを使用する	57
5-11. JavaScript を外部エディタを用いて編集する	59
任意の外部エディタを登録する	59
外部エディタを起動する	60
6. 制限事項・変更点	61

6-1. 制限事項	61
オブジェクト設定上の制限	61
JavaScript に関する制限	62
アクションに関する制限	62
数値フォーマットの制限	62
6-2. 注意事項	63
改ページ・マルチフォーム・セット出力に関して	63
PDF ファイル出力時の JavaScript の動作について	64
ビューアアプリケーション上での動作について	64

1. PDF フォーム機能の概要

1-1. PDF フォーム機能の概要

PDF フォームは PDF ファイル上に設けられたフィールドに対してテキスト入力や選択、電子署名といったユーザと PDF ファイル間の対話を可能にするインタラクティブ（対話）機能です。Create! FormCollect では、PDF フォーム機能を実装するためのオブジェクト（以下フィールドオブジェクトと呼びます）を含む PDF ファイルの出力を行うことができます。

PDF フォーム機能は Adobe Acrobat、もしくは Adobe Reader 等の PDF ビューア上で動作します。以降の説明では便宜上これらのアプリケーションを『ビューアアプリケーション』と呼ぶこととします。

1-2. Create! FormCollect が提供する主な機能

PDF フォーム機能

Create! FormCollect では 9 種類のフィールドオブジェクトをサポートしています。

テキストフィールド	ボタンフィールド
コンボボックスフィールド	サブミットボタンフィールド
リストボックスフィールド	リセットフォームボタンフィールド
ラジオボタンフィールド	署名フィールド
チェックボックスフィールド	

これらのオブジェクトの機能を用いて以下のような機能を含んだ PDF ファイルを出力することができます。

- フィールドに対するテキスト入力
- フィールド項目の選択
- サーバスクリプトへのフォームデータ送信
- フィールド間の計算
- 入力したテキストの表示形式の変換
- フィールドデータの数値範囲の検証
- フィールド操作によるアクションの実行

アクション機能

Create! FormCollect では、フィールドオブジェクトを用いたインタラクティブな PDF 機能のほかにフィールドデータを WEB サーバに送信する、入力した内容をクリアする、JavaScript を組み込んでビューアアプリケーション上で任意の動作を行わせるといったアクション機能を組み込むことができます。

アクション機能の詳細については、「3-5. アクションを組み込む」をご覧ください。

2. フィールドの作成

2-1. フィールドを作る

フィールドオブジェクトの定義方法

フィールドオブジェクトはFormDesign 付属の Form エディタ上で定義します。Form エディタのオブジェクトツールバーから各フィールドオブジェクトを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで定義されます。

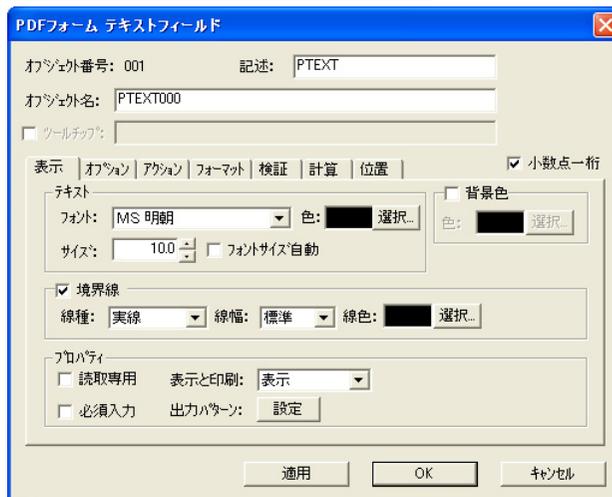
図：オブジェクトツールバー



フィールドオブジェクト属性ダイアログの構成

定義されたオブジェクトをダブルクリックすることでフィールドオブジェクト属性ダイアログが表示されます。

図：フィールドオブジェクト属性ダイアログ



フィールドオブジェクト属性ダイアログは次のようなセクションから構成されています。

[表示]セクション

フィールドオブジェクトの外観を設定します。

[オプション]セクション

各フィールド特有の設定を行います。

[アクション]セクション

フィールドオブジェクトに対する操作に応じて動作するアクションを設定します。
※このセクションはサブミットボタン、リセットフォームボタンには含まれません。

[フォーマット]セクション

フィールドオブジェクトに表示されるテキスト文字列の表示形式を設定します。
※このセクションはテキストフィールド、コンボボックスフィールドのみに含まれます。

[検証]セクション

フィールドオブジェクトの値の範囲等を検証します。
※このセクションはテキストフィールド、コンボボックスフィールドのみに含まれます。

[計算]セクション

他のフィールドオブジェクト間の値の計算方法を設定します。
※このセクションはテキストフィールド、コンボボックスフィールドのみに含まれます。

[署名後の動作]セクション

署名後の PDF ファイル上の制御を設定します。
※このセクションは署名フィールドのみに含まれます。

[選択の変更]セクション

項目の選択時の PDF ファイル上の制御を設定します。
※このセクションはリストボックスフィールドのみに含まれます。

[位置]セクション

フィールドオブジェクトの定義位置を設定します。

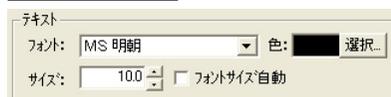
2-2. フィールドの外観、表示状態を設定する

各フィールドオブジェクト属性ダイアログ上の [表示] セクションでは、フィールドの外観（見た目）に関する設定を行うことができます。

フォントを設定する

フィールドオブジェクト属性ダイアログの [表示]-[テキスト] でフィールドオブジェクト内で表示されるテキスト文字列のフォント設定を行うことができます。

図：テキスト設定



[フォント]

フィールドオブジェクト上で表示されるテキスト文字列のフォントを指定します。
フィールドオブジェクトで設定可能なフォント種別は以下のとおりです。

[欧文書体]		[日本語書体]
Courier	Times-Roman	MS 明朝
Courier-Bold	Times-Bold	MS ゴシック
Courier-Oblique	Times-Italic	
Courier-BoldOblique	Times-BoldItalic	
Helvetica	ZapfDingbats	
Helvetica-Oblique	Symbol	
Helvetica-Bold		
Helvetica-BoldOblique		

※ラジオボタンフィールド、チェックボックスフィールドでは固定で“ZapfDingbats”になります。

[サイズ]

フィールドオブジェクト上で表示されるテキスト文字列のフォントサイズを指定します。フォントサイズには2～300までの値を指定することができます。

[フォントサイズ自動]を指定すると、入力した文字列の長さによってフォントサイズが4pt～144ptの間で自動的に調整されるようになります。

[色]

フィールドオブジェクト上で表示されるテキスト文字列の色を設定します。

背景色を設定する

フィールドオブジェクト属性ダイアログの[表示]-[背景色]でフィールドオブジェクトの領域を塗りつぶす任意の背景色を指定することができます。

図：背景色設定



[背景色]

フィールドオブジェクト領域内の背景色の有無を設定します。

[色]

フィールドオブジェクト領域内を塗りつぶす色を設定します。

境界線を設定する

フィールドオブジェクト属性ダイアログの[表示]-[境界線]でフィールドオブジェクトの領域を囲む境界線の有無やスタイルを設定することができます。

図：境界線設定



[境界線]

フィールドオブジェクトの領域を囲む境界線の描画の有無を設定します。

[スタイル]

フィールドオブジェクトの領域を囲む境界線のスタイルを指定します。設定可能な境界線スタイルは、

実線 破線 ベベル 切り込み 下線

の5種類です。

図：境界線スタイルの出力例

**[線幅]**

フィールドオブジェクトの領域を囲む境界線の線幅を“細”、“標準”、“太”から指定します。

[線色]

フィールドオブジェクトの領域を囲む境界線の色を指定します。

フィールドの表示、印刷時の状態を設定する

フィールドオブジェクト属性ダイアログの [表示]-[プロパティ]-[表示と印刷] では、出力された PDF ファイルをビューアアプリケーション上で表示、もしくは印刷した時のフィールドオブジェクトの表示状態を指定することができます。設定項目の内容は以下の表の通りです。

図：[表示と印刷] 設定



表：[表示と印刷] 設定内容

選択項目	ビューア上での表示	印刷時の表示
表示	○	○
非表示	×	×
表示 / 印刷しない	○	×
非表示 / 印刷する	×	○

フィールドを読取専用にする

フィールドオブジェクト属性ダイアログの [表示]-[プロパティ]-[読取専用] を設定することで、ビューアアプリケーション上でフィールドオブジェクトを操作することができなくなります。

図：読取専用設定

フィールドへの入力（選択）を必須にする

フィールドオブジェクト属性ダイアログの [表示]-[プロパティ]-[必須入力] を設定することで、サブミットフォームでフィールドデータをサーバスクリプトへ送信する際に、ビューアアプリケーションによってフィールドの空白データ（データの有無）の検知が行われます。サブミットフォームについては、「5-3. 指定したサーバスクリプトにフィールドデータを送信する」をご覧ください。

図：必須設定

2-3. テキスト入力を行うフィールドを作る - テキストフィールド

テキストフィールドは、ビューアアプリケーション上の PDF ファイルに対してテキスト入力を行うことができます。

定義方法

Form エディタのオブジェクトツールバーからテキストフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このテキストフィールドに対して文字数制限などの特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：テキストフィールドオプション設定

各オプションの詳細については以下の項目を参照してください。

初期値を指定する

テキストフィールドオプション設定の [初期値] では、リセットフォームが行われた場合やデータマッピングされていない場合にテキストフィールド内に表示されるテキスト文字列を指定します。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

初期値には最大 9,999 文字の文字列を指定することができます。
また、[複数行を許可する] を指定すると、改行を含む複数行のテキストを初期値として指定することが可能です。

テキストの配置を指定する

PDF ファイルの出力時、もしくはビューア上での入力時のテキストフィールド領域内でのテキスト文字列の表示位置を指定します。

この設定を行うには、テキストフィールドオプション設定の [配置] で右揃え、中央揃え、左揃えから選択します。初期状態では左寄せが指定されています。

なお [パスワード] が設定されている場合には、この設定を行うことはできません。この場合は左揃えが指定されます。

図：配置によるテキスト表示位置



入力可能な最大文字数を指定する

テキストフィールド内に入力可能な最大文字数を指定します。

この設定を行うには、テキストフィールドオプション設定で [最大文字数の指定] を選択して任意の最大文字数を指定します。この値はバイト数ではなく文字数となります。最大文字数には 0 ~ 10,240 文字を指定することができます。

※ PDF ファイル上のテキストフィールドには最大 10,240 文字を直接入力可能ですが、データマッピングから指定可能な最大値は 10,240 バイトとなります。全て全角文字を指定した場合、最大 5,120 文字の指定が可能です。

なお [マス目で区切る] が設定されている場合には、この設定を行うことはできません。

複数行のテキスト入力を可能にする

テキストフィールドに対して複数行のテキスト入力を可能にします。

この設定を行うには、テキストフィールドオプション設定で [複数行を許可する] を選択します。

なお [パスワード]、[マス目で区切る] が設定されている場合には、この設定を行うことはできません。

フィールド内でスクロールを可能にする

テキストフィールド内へ入力したテキストの長さが領域を越えてしまった場合に、テキストをすべて確認することができるようにスクロールを有効にします。

この設定を行うには、テキストフィールドオプション設定で [スクロールを許可する] を選択します。

なお [マス目で区切る] が設定されている場合には、この設定を行うことはできません。

パスワード入力を行うフィールドを作る

テキストフィールド内に入力した文字列を表記上アスタリスク (*) として表示させます。

この設定を行うには、テキストフィールドオプション設定で [パスワード] を選択します。

図：[パスワード]出力例



なお [複数行を許可する]、[マス目で区切る]、[スペルチェック] のいずれかが設定されている場合には、この設定を行うことはできません。

※このオプションは入力したテキストをパスワードとしてビューア上ではアスタリスク (*) 表示しますが、PDF ファイル内部では入力された値がそのまま保持されます。

マス目に区切られたフィールドを作る

テキストフィールドの横領域を指定した文字数で分割し、マス目ごとに文字を表示することを可能にします。

この設定を行うには、テキストフィールドオプション設定で [マス目で区切る] を選択し、任意の文字数を指定します。この値はバイト数ではなく文字数となります。文字数は 0 ~ 999 文字を指定することができます。

図：[マス目で区切る]出力例



なお [複数行を許可する]、[スクロールを許可する]、[最大文字数の指定]、[パスワード]、[スペルチェック] が設定されている場合には、この設定を行うことはできません。

入力時にスペルチェックを行う

テキストフィールド内に入力を行った際にビューアに設定されている辞書に基づき、入力テキストのスペルチェックを行うことを可能にします。

この設定を行うには、テキストフィールドオプション設定で [スペルチェック] を選択します。

なお [パスワード]、[マス目で区切る] が設定されている場合には、この設定を行うことはできません。

※スペルチェックの基になる辞書の指定については、Adobe Acrobat、Adobe Reader 等のビューアの設定をご覧ください。

出力時にフィールドに出力するデータが空白の場合、初期値を出力する

PDF ファイルの出力時にデータマップによって取得した外部データが空白の場合に、初期値で指定したテキスト文字列を表示するかどうかを指定することができます。

この設定を行うには、テキストフィールドオプション設定で [空白時に初期値を使用] を選択します。

※この設定は Create! FormCollect によって PDF ファイルを出力した初期状態でのみ有効です。出力したファイルをビューア上で編集した際には無効となりますのでご注意ください。

2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド

コンボボックスフィールドは PDF ファイル上に項目の一覧をポップアップ表示します。

定義方法

Form エディタのオブジェクトツールバーからコンボボックスフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。このコンボボックスフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：コンボボックスフィールドオプション設定

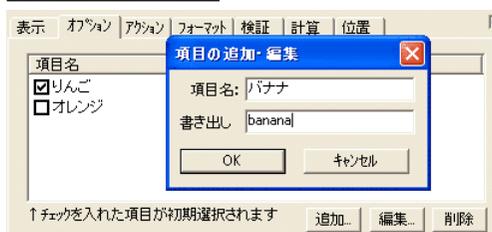


項目を追加（編集・削除）する

メニュー項目を追加する

メニューに表示する項目を登録するためには、[追加] ボタンを押下して表示される [項目の追加・編集] ダイアログを用いて行います。メニュー項目は256個まで登録することができます。

図：項目の追加・編集



[項目名]

実際に PDF ファイル上でメニューの選択肢として表示されるテキストです。

[書き出し値]

サブミットフォーム時やフォームデータの書き出し時に選択された項目の値として出力される値です。この値を設定しない場合は、項目名が代替値として出力されます。

メニュー項目を変更する

メニュー項目の一覧で選択されている項目を編集するためには、[編集] ボタンを押下して表示される項目の追加・編集ダイアログを用いて変更します。

メニュー項目を削除する

メニュー項目の一覧で選択されている項目を削除するためには、[削除] ボタンを押下します。また一覧の項目をすべて削除したい場合には、[全て削除] ボタンを押下します。

外部の CSV ファイルから項目をインポートする

メニュー項目が多数存在する場合、一つ一つ手作業で登録する手間が発生します。この手間を省くために外部の CSV ファイルにあらかじめメニュー項目を登録しておき、それをフォーム作成時にインポートすることで項目登録作業を簡略化することができます。

外部 CSV ファイルからメニュー項目をインポートするためには、[インポート] ボタンを押下して表示される CSV データインポートダイアログを使用します。

図：[CSV データインポート]



[ファイル指定]

[参照] ボタンを押下してメニュー項目のインポートに使用する CSV ファイルを指定します。

[パターン]

メニュー項目のインポートに使用する CSV ファイルのデータパターンを選択します。Create! FormCollect では、次の 3 パターンの CSV データからメニュー項目をインポートすることができます。

A パターン：1 行に 1 項目の項目名と書き出し値を記述したデータ。

[例] name1, value1,
name2, value2,
name3, value3,
.....

B パターン：項目名を 1 行目に、書き出し値を 2 行目にそれぞれ区切り文字で区切って繰り返したデータ。

[例] name1, name2, name3,
value1, value2, value3,
.....

C パターン：項目名と書き出し値を区切り文字ごとに繰り返したデータ。

[例] name1, value1, name2, value2, name3, value3,
.....

※上記の例では区切り文字にカンマを使用しています。

また項目名のみを記述した CSV ファイルを使用する場合には、[項目名を書き出し値と併用する]を設定します。この場合、項目名のみがメニュー項目の一覧として登録されます。

[CSV データ設定]

CSV ファイル内で使用されている区切り文字と制御文字を任意に指定することができます。区切り文字、制御文字にはアスキーコード 0x21 ~ 0x3F 内の半角数字および半角記号を指定します。

初期の状態では区切り文字にカンマ (,)、制御文字にダブルクォーテーション (") が指定されています。

初期選択項目を指定する

登録項目の一覧上で項目名の左横にあるチェックボックスを選択することにより、このメニュー項目の初期選択項目を設定することができます。

リセットフォームが行われた場合やデータマッピングされていない場合には、ここで指定した項目が選択された状態になります。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

図：初期選択項目

項目名	書き出し値
<input checked="" type="checkbox"/> りんご	apple
<input type="checkbox"/> オレンジ	orange
<input type="checkbox"/> バナナ	banana

登録した項目を並べ替える

PDF ファイル出力時に登録されたメニュー項目を数字、アルファベット順に並べ替えることができます。

この設定を行うには、コンボボックスフィールドオプション設定で [項目の並べ替え] を選択します。

※この設定は、PDF ファイル出力時に外部データからメニュー項目をインポートする際にも有効となります。

項目を編集可能にする

メニュー内に適当な選択項目が存在しない場合などに、テキストフィールドと同様にコンボボックス上にテキスト文字列を入力することができます。この場合、メニュー項目以外の値をフィールド値として扱うことができます。

この設定を行うには、コンボボックスフィールドオプション設定で [編集可能] を選択します。

項目のスペルチェックを行う

コンボボックスフィールドの選択項目に対して、ビューアに設定されている辞書に基づいたスペルチェックを行うことができます。

この設定を行うには、コンボボックスフィールドオプション設定で [スペルチェック] を選択します。

※スペルチェックの基になる辞書の指定については、Adobe Acrobat、Adobe Reader 等のビューアの設定をご覧ください。

2-5. メニューリストを作る - リストボックスフィールド

リストボックスフィールドは PDF ファイル上に項目の一覧を表示します。登録された項目が領域内に収まらない場合には、自動的にスクロールバーがつけられます。

定義方法

Form エディタのオブジェクトツールバーからリストボックスフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで定義されます。

このリストボックスフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：リストボックスフィールドオプション設定



複数の項目を選択可能にする

リストボックスフィールドは通常単一の項目のみの選択となりますが、オプションを指定することにより複数の項目をビューアアプリケーション上で選択することができるようになります。

この設定を行うには、リストボックスフィールドオプション設定で [複数選択] を選択します。

項目を追加（編集・削除）する

※前述の「2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド」の項をご覧ください。

外部の CSV ファイルから項目をインポートする

※前述の「2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド」の項をご覧ください。

初期選択項目を指定する

※前述の「2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド」の項をご覧ください。

登録した項目を並べ替える

※前述の「2-4. 単一項目が選択可能なメニューを作る - コンボボックスフィールド」の項をご覧ください。

2-6. ラジオボタンを作る - ラジオボタンフィールド

ラジオボタンフィールドは複数の選択項目のうち1つだけ選択する形式のボタンです。

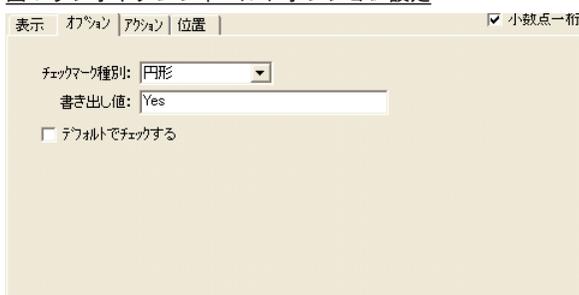
定義方法

Formエディタのオブジェクトツールバーからラジオボタンフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

※共通の項目に対する選択肢としてグループ化するラジオボタンフィールドは、全て同じオブジェクト名、異なる書き出し値を設定する必要があります。

このラジオボタンフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の「オプション」セクションで設定を行います。

図：ラジオボタンフィールドオプション設定



チェックマークのスタイルを指定する

選択した際のチェックマークの表示形式を指定することができます。

この設定を行うには、ラジオボタンフィールドオプション設定の「チェックマーク種別」から選択します。チェックマーク種別は「チェック」、「ひし形」、「円形」、「四角形」、「十字形」、「星形」から選択することができます。

図：チェックマーク種別の出力例



選択時の値を指定する

サブミットフォームなどフォームデータの書き出しの際に出力されるフィールドの実値データを設定することができます。

この設定を行うためには、ラジオボタンフィールドオプション設定の「書き出し値」に値を設定します。

デフォルトで選択状態にする

リセットフォームが行われた場合やデータマッピングされていない場合に選択状態にする初期設定を行うことができます。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

この設定を行うためには、ラジオボタンフィールドオプション設定の [デフォルトでチェックする] を選択します。

2-7. チェックボックスを作る - チェックボックスフィールド

チェックボックスフィールドは、該当する項目の可否を選択する形式のボタンです。

定義方法

Form エディタのオブジェクトツールバーからチェックボタンフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このチェックボックスフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：チェックボックスフィールドオプション設定

表示 オプション | アクション | 位置 小数点一桁

チェックマーク種別:

書き出し値:

デフォルトでチェックする

チェックマークのスタイルを指定する

※前述の「2-6. ラジオボタンを作る - ラジオボタンフィールド」の項をご覧ください。

チェック時の値を指定する

※前述の「2-6. ラジオボタンを作る - ラジオボタンフィールド」の項をご覧ください。

デフォルトで選択状態にする

※前述の「2-6. ラジオボタンを作る - ラジオボタンフィールド」の項をご覧ください。

2-8. ボタンを作る - ボタンフィールド

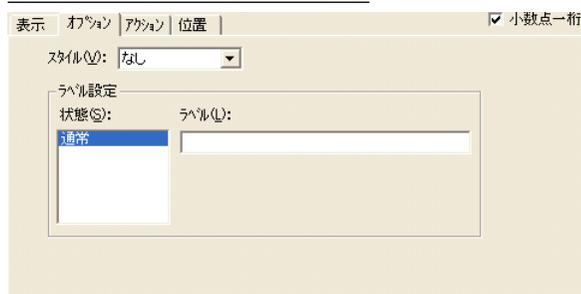
ボタンフィールドはアクションを動作させるトリガーの役割として利用することができます。

定義方法

Form エディタのオブジェクトツールバーからボタンフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このボタンフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：ボタンフィールドオプション設定



ボタンに表示するテキストを指定する

[スタイル]

ボタンフィールドを押下した際のボタンの表示方法を指定することができます。

押下時のスタイルは、

- なし：外観は変わりません。
- 反転：押下時にボタンフィールド領域内の色が反転します。
- アウトライン：押下時にボタンフィールド領域枠が強調されます。
- プッシュ：押下時にボタンフィールドが押し込まれたような外観に変わります。

から選択することができます。

[ラベル設定]

ボタンフィールドに表示されるテキスト文字列を設定します。

マウスの状態 (“通常”、“押下”、“ロールオーバー”) ごとにボタンのラベル表示を変えるには、ボタンの押下時のスタイルで “プッシュ” を選択します。

※その他のスタイルを選択している場合は、“通常” のラベル文字列のみ指定ができます。

各マウスの状態は、

- 通常：マウスをクリックしていない時のボタンフィールドの表示方法
- 押下：マウスをクリックして放さない間のボタンフィールドの表示方法
- ロールオーバー：マウスポインタがボタンフィールドの上にある時の表示方法

となります。押下時のスタイルで “プッシュ” を選択をしている場合には、それぞれのマウスの状態に対してラベル文字列を設定します。

図：出力例



2-9. 送信ボタンを作る - サブミットボタンフィールド

サブミットボタンフィールドは、Web サーバへフィールドデータを送信する目的に特化したボタンフィールドです。データの送信については、「5-3. 指定したサーバスクリプトにフィールドデータを送信する」をご覧ください。

定義方法

Form エディタのオブジェクトツールバーからサブミットボタンフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。このサブミットボタンフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：サブミットボタンフィールドオプション設定

表示 | オプション | 位置 | 小観点一桁

表示するラベル:

送信先 URL:

データ送信形式

- FDF (F)
- HTML (H)
- XML (XML)
- 文書全体 (PDF) (P)

対象のフィールド

- 全てのフィールド (A)
- フィールドの指定 (S)

ボタンに表示するテキストを指定する

サブミットボタンフィールドに表示されるテキスト文字列を設定することができます。

ラベル文字列を設定するには、サブミットボタンフィールドオプション設定の [表示するラベル] にテキスト文字列を指定します。この文字列はボタンフィールド上で中央揃えで表示されます。

ラベル文字列には最大 99 文字のテキスト文字列を指定できます。

フィールドデータを処理するサーバスクリプトの URL を指定する

フィールドデータ送信先のサーバスクリプトの URL を指定することができます。

サーバスクリプトでは送信されたフィールドデータを処理する仕組みを持っている必要があります。

送信先 URL を設定するには、サブミットボタンフィールドオプション設定の [送信先 URL] に URL 文字列を指定します。

送信時のデータ形式を指定する

フィールドデータを送信する際のデータ形式を指定することができます。
データ送信形式は、

HTML FDF XML (XFDF) PDF

からサブミットボタンフィールドオプション設定の [データ送信形式] で指定します。

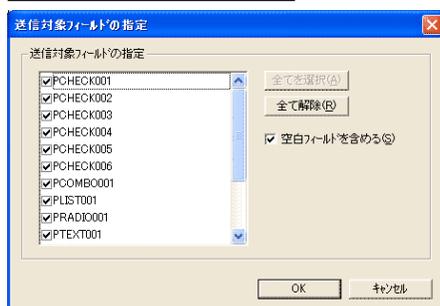
各データ形式の詳細については、「5-3. 指定したサーバスクリプトにフィールドデータを送信する」をご覧ください。

データを送信する対象のフィールドを指定する

サーバスクリプトにフィールドデータを送信するフィールドを指定することができます。
この設定を行うには、サブミットボタンフィールドオプション設定の [対象のフィールド] を設定します。全てのフィールドの値を送信する場合には [すべて] を、指定した任意のフィールドの値を送信する場合には [フィールドの指定] を選択します。

[フィールドの指定] を選択した場合は、更に [指定のフィールド] ボタンを押下して [送信フィールドの指定] ダイアログ上で送信するフィールドを指定します。

図 : [送信フィールドの指定]



[送信対象のフィールドの指定]

フィールド一覧には送信対象となるフィールド名が表示されています。その左横のチェックボックスではフィールドを送信対象とするかどうかの設定を行います。

[空白フィールドを含む]

この設定が行われていない場合、送信する際にフィールドデータが空のフィールドはサーバスクリプトには値が送信されません。設定されている場合には空のデータが送信されます。

2-10. クリアボタンを作る - リセットフォームボタンフィールド

リセットフォームボタンフィールドは、ビューアプリケーション上で入力、選択されたフィールドデータを各フィールドの初期値にリセットする機能に特化したボタンフィールドです。

定義方法

Form エディタのオブジェクトツールバーからリセットフォームボタンフィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このリセットフォームボタンフィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [オプション] セクションで設定を行います。

図：リセットフォームボタンフィールドオプション設定



ボタンに表示するテキストを指定する

※「2-9. 送信ボタンを作る - サブミットボタンフィールド」をご覧ください。

フィールドデータの内容をリセットする対象のフィールドを指定する

フィールドデータをリセットするフィールドを指定することができます。

この設定を行うには、リセットフォームボタンフィールドオプション設定の [対象のフィールド] を設定します。全てのフィールドの値をリセットする場合には [すべて] を、指定した任意のフィールドの値をリセットする場合には [フィールドの指定] を選択します。

[フィールドの指定] を選択した場合は、更に [指定のフィールド] ボタンを押下して [リセット対象フィールドの指定] ダイアログ上でリセットするフィールドを指定します。

図：[リセット対象フィールドの指定]



[リセット対象フィールドの指定]

フィールド一覧にはリセット対象となるフィールド名が表示されています。その左横のチェックボックスではフィールドをリセットの対象とするかどうかの設定を行います。

2-11. 電子署名を行う領域を作る - 署名フィールド

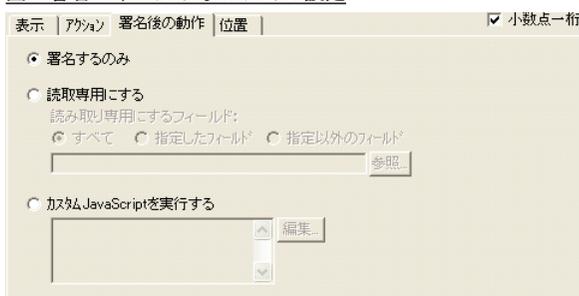
署名フィールドは、ユーザがビューアプリケーション上で電子署名を付加する目的に特化した領域です。ビューアプリケーション上でこの領域をクリックすると電子署名のプロセスが開始されます。

定義方法

Formエディタのオブジェクトツールバーから署名フィールドを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

この署名フィールドに対する特有の設定を行うためには、フィールドオブジェクト属性ダイアログ内の [署名後の動作] セクションで設定を行います。このセクションは、ビューアプリケーション上でこのフィールドを用いて電子署名を行った際の動作を設定します。

図：署名フィールドオプション設定



署名を行った後にフィールドを読取専用にする

署名フィールドに対して電子署名を行った際に任意のフィールドを読取専用にすることができます。

この設定を行うには、[署名後の動作] セクションの [読取専用にする] を選択します。任意のフィールドを指定する場合には、[指定のフィールド]、もしくは [指定以外のフィールド] を選択し、[参照] ボタンから対象とするフィールドを選択します。

図：[フィールドの選択]



[全フィールド]

署名後に読取専用にすることが可能な全フィールド名が一覧表示されています。

[選択されているフィールド]

署名後に読取専用にする対象のフィールド名が一覧表示されています。

[追加]

[全フィールド] 一覧で選択したフィールドを [選択されているフィールド] 一覧に移動し、対象のフィールドとして選択します。

[削除]

[選択されているフィールド] 一覧で選択したフィールドを [全フィールド] 一覧に移動し、選択を解除します。

署名を行った後に任意の動作を行わせる

署名フィールドに対して電子署名を行った際にカスタム JavaScript を用いて任意の動作を PDF 上で行わせることができます。

この設定を行うには、[署名後の動作] セクションの [カスタム JavaScript を実行する] を選択して任意の JavaScript コードを設定します。

カスタム JavaScript の設定方法は、「5-7. JavaScript を使用する」をご覧ください。

3. フィールドに特殊機能をつける

3-1. 入力したテキストの表示形式を指定する - フォーマット

テキストフィールド、コンボボックスフィールドの属性ダイアログに属する [フォーマット] セクションでは、フィールドに表示されるデータ形式を設定することができます。

フォーマットには“数値”、“パーセント”、“日付”、“カスタム”からデータ形式を指定することができます。

“数値”、“パーセント”、“日付”のいずれかのフォーマットを選択することにより以下の項目で説明する詳細なフォーマット設定を行うことができます。

入力したテキストを数値表記する

[フォーマット] セクションの [分類] から “数値” を選択することでフィールド内に入力することができる値を数値のみに設定することができます。

また、[小数点以下の桁数]、[通貨記号]、[桁区切りの表記方法]、[負数の表記方法] といった設定を行うことにより入力したテキストに対して表示形式を指定することができます。

図：数値フォーマット設定

[小数点以下の桁数]

入力されたデータのフィールドでの小数点以下の表示桁数を指定します。設定可能な桁数は 0 ～ 3 桁までとなります。

図：小数点以下の桁数 - 「1234.5678」を入力した場合

桁数	出力例	桁数	出力例
0桁	1,235	2桁	1,234.57
1桁	1,234.6	3桁	1,234.568

[通貨記号]

入力されたデータに対して指定した通貨記号をつけた形式でフィールド上に表示します。
設定可能な通貨記号は、

円 (¥) ドル (\$) ポンド (£)
フラン (F) ユーロ (EUR) マルク (DM)

となります。

図：通貨記号 - 「1234」を入力した場合

通貨記号	出力例	通貨記号	出力例
円 (¥)	¥1,234	フラン (F)	1,234 F
ドル (\$)	\$1,234	ユーロ (EUR)	EUR1,234
ポンド (£)	£1,234	マルク (DM)	1,234 DM

[桁区切りの表記方法]

入力されたデータを指定の桁区切り方法を用いてフィールド上に表示します。

図：桁区切りの表記方法 - 「1234.567(1234,567)」を入力した場合

■小数点が「.」の場合

■小数点が「,」の場合

種別	出力例	種別	出力例
1,234.56	1,234.567	1.234,56	1.234,567
1234.56	1234.567	1234,56	1234,567

[負数の表記方法]

入力されたデータが負数の場合、指定した表記方法でフィールドに表示します。

図：負数の表記方法 - 「-1234」を入力した場合

種別	出力例	種別	出力例
マイナス記号	-1,234	括弧囲み表記	(1,234)
朱記表記	1,234	カッコ囲み 朱記表記	(1,234)

入力したテキストをパーセント表記する

[フォーマット] セクションの [分類] から “パーセント” を選択することでフィールド内に
入力することができる値を数値のみに設定することができます。パーセントフォーマットを設
定した場合、フィールド上の表示は “入力した数値データ × 100” の値にパーセントマーク (%)
がついた形式となります。例えば、フィールドに “0.5” と入力した場合にはフィールド上では
“50%” と表示されます。

図：パーセントフォーマット設定

【小数点以下の桁数】

※前述の『入力したテキストを数値表記する』をご覧ください。

【桁区切の表記方法】

※前述の『入力したテキストを数値表記する』をご覧ください。

入力したテキストを日付表記する

[フォーマット] セクションの [分類] から "日付" を選択することでフィールド内に入力した日付データを指定した日付形式で表示することができます。

図：日付フォーマット設定

【日付の表記方法】

入力された日付を表すデータを指定した日付形式でフィールド上に表示します。

図：日付の表記方法 - 「2005/9/1」を表す場合

種別	入力例	出力例
yyyy年mm月dd日	2005/9/1	2005年09月01日
yyyy-mm-dd	2005/9/1	2005-09-01
mm/dd/yyyy	9/1/2005	09/01/2005
mmm d, yyyy	9/1/2005	September 1, 2005

カスタム JavaScript を使用して独自の表示形式を指定する

[書式カスタム JavaScript]

書式カスタム JavaScript を設定することで、フィールド上に入力したデータを独自の表示形式で表示することができます。例えば、独自の通貨記号を入力データに付加する場合や、入力した商品コードを実際の商品名で表示することができます。

[キーストロークカスタム JavaScript]

キーストロークとは、“キーのひと打ち”を意味します。

このキーストロークカスタム JavaScript を設定することで、フィールド上でキーを打った際に任意の動作を行わせることができます。例えば、フィールド上に入力できるテキスト文字を特定の文字に制限することができます。

カスタム JavaScript の設定方法は、「5-7. JavaScript を使用する」をご覧ください。

※カスタム JavaScript を使用したフォーマット設定は、ビューアプリケーション上でのみ動作可能となります。Create! FormCollect からの PDF ファイル出力時には表示形式は反映されませんのでご注意ください。

3-2. 他のフィールド間の値の計算結果を表示させる - 計算

テキストフィールド、コンボボックスフィールドの属性ダイアログに属する [計算] セクションでは、他のフィールドの値を用いた計算設定を行うことができます。この設定は、フォーマット設定において“数値”、“パーセント”フォーマットが設定されている場合のみ有効となります。

図：計算設定



フィールドを指定して計算に使用する

テキストフィールド、コンボボックスフィールドには他のフィールドの値を用いた計算結果を表示することができます。

計算の方法、使用するフィールドの指定は [計算] セクションの項目で設定します。

[計算の種類]

計算方法を設定するには、[計算の種類] で、

和 差 積 商 平均 最小 最大 カスタム

の中から選択します。

[計算に使うカラム]

[計算の種類] で選択した計算方法で計算させるフィールドを指定します。

[編集] を押下すると計算に使用するフィールドの選択ダイアログが表示されます。

図：計算に使用するフィールドの選択ダイアログ



左側には計算に使用することができるフィールド名が一覧になっています。計算に使用したフィールド名を選択して [追加] をクリックすることで右側の計算対象フィールドの一覧に追加されます。また、計算対象からフィールドを削除したい場合は、右側の一覧から対象のフィールド名を選択して [削除] をクリックすることで、左側の一覧にフィールド名を戻します。

※ [計算に使うカラム] で選択したフィールドの順番は計算の順序に影響します。例えば、[計算の種類] で “差” を選択した場合、[計算に使用するカラム] において “VALUE1 (値 =5)”、“VALUE2 (値 =3)”、“VALUE3 (値 =1)” の順で追加した場合には計算は次のように行われます。

$$\text{VALUE1 (5)} - \text{VALUE2 (3)} - \text{VALUE3 (1)} = 1$$

この追加の順番を逆から行くと、

$$\text{VALUE3 (1)} - \text{VALUE2 (3)} - \text{VALUE1 (5)} = -7$$

となります。つまり “差”、“商” といった計算方法が指定されている場合には、フィールドの追加する順番によって計算結果が異なる可能性がありますのでご注意ください。

また、フォーム上に計算が設定されたフィールドが複数定義されている場合には、その計算順序を指定することができます。この詳細については「5-6. 計算を行う順序を指定する」をご覧ください。

任意の値を計算に使用する

前述のフィールド間の計算結果に対して、固定の値を用いた計算を設定することができます。

固定値を用いた計算方法は [固定値]-[計算の種類] で、

和 差 積 商

から選択することができます。

固定値を用いた計算は、次のような消費税込みの金額を算出したい場合などで有効です。

計算例：VALUE1、VALUE2 の値の合計に 1.05 を掛けた消費税込みの金額を VALUE3 に表示する

※この計算設定は、計算結果を表示する VALUE3 で行います。

① [計算の種類] で “和” を選択する。

計算の種類:

② [計算で使用するカラム] で “VALUE1”、“VALUE2” を選択する。

計算に使うカラム:

③ [固定値] の [計算の種類] で “積” を選択する。

固定値
計算の種類:

④ [固定値] で “1.05” を指定する。

固定値:

以上の設定を行うことで VALUE3 には VALUE1、VALUE2 の値の合計に消費税を含んだ金額が表示されます。

VALUE1	+	VALUE2	=	VALUE3
¥1,000		¥500		¥1,575

※固定値を用いた計算は、フィールド間の計算結果に対して指定の値を計算させるものです。指定の値に対してフィールド間の計算結果の値を計算させることはできません。このような計算を行いたい場合には、後述の「カスタム JavaScript を使用して独自の計算方法を指定する」をご覧ください。

カスタム JavaScript を使用して独自の計算方法を指定する

カスタム JavaScript を用いて独自の計算方法を作成することができます。

例えば、四則演算と組み合わせた複雑な計算方法を埋め込むこともできます。

この設定を行うには、[計算] セクションの [カスタム JavaScript を実行する] を選択して任意の JavaScript コードを設定します。

カスタム JavaScript の設定方法は、「5-7. JavaScript を使用する」をご覧ください。

※カスタム JavaScript を使用した計算方法は、ビューアプリケーション上でのみ動作可能となります。PDF ファイル出力時には計算結果が反映されませんのでご注意ください。

3-3. 入力した（選択した）フィールドの値を検証する - 検証

テキストフィールド、コンボボックスフィールドの [検証] セクションではフィールドに入力できる値を制限するための設定を行うことができます。この設定により、フィールド上に適切なデータのみを入力させることができるようになります。

図：検証設定

フィールドの値の範囲を数値で指定する

数値、パーセントフォーマットが設定されている場合、フィールド内に入力可能な最小値、最大値を指定することができます。

この設定を行うには、[検証] セクションの [値の範囲を指定する] を選択します。最小値を検証する場合には、[最小値] の左のチェックボックスを選択して任意の最小値を指定します。同様に、最大値を検証する場合には、[最大値] の左のチェックボックスを選択して任意の最大値を指定します。

図：値の範囲設定

設定可能な数値の範囲は、-9,999,999,999.999 ~ 9,999,999,999.999 までの値となります。

カスタム JavaScript を使用して独自の検証方法を指定する

カスタム JavaScript を使用することで入力データに対する独自の検証を行わせることができます。例えば、入力データを英数字に限定するといった制御を埋め込むこともできます。この設定を行うには、[検証] セクションの [カスタム JavaScript を実行する] を選択して任意の JavaScript コードを設定します。

カスタム JavaScript の設定方法は、「5-7. JavaScript を使用する」をご覧ください。

※カスタム JavaScript を使用した入力データの検証は、ビューアアプリケーション上でのみ動作可能となります。PDF ファイル出力時にはスクリプトの内容が反映されませんのでご注意ください。

3-4. メニュー項目の選択の変更時に任意の処理を行う

カスタム JavaScript を使用して選択項目の変更時の動作を指定する

ビューアアプリケーション上でリストボックスの項目を変更した際にカスタム JavaScript を用いて任意の動作を PDF ファイル上で行わせることができます。

この設定を行うには、[選択の変更] セクションの [カスタム JavaScript を実行する] を選択して任意の JavaScript コードを設定します。

カスタム JavaScript の設定方法は、「5-7. JavaScript を使用する」をご覧ください。

3-5. アクションを組み込む

アクションとは、PDF ファイルの静的な表現に加え、ページが切り替った際の動作やフィールドに対する操作が行われた際の動作などを設定することができる機能です。このアクションを利用することで PDF ファイルに対して動きや対話性を高めることができます。

Create! FormCollect では、以下の項目に対してアクションを設定することができます。

ページ

ページを開いた時

ページを閉じた時

フィールド

マウスボタンを放した時

マウスボタンを押した時

マウスポインタをフィールド領域内に合わせた時

マウスポインタをフィールド領域外に出した時

フォーカスを合わせた時

フォーカスを外した時

これらの項目はアクションを起こす“きっかけ”という意味から、以降『トリガー』と呼ぶこととします。

これらのトリガーに対して次のアクションを設定することができます。

JavaScript の実行

フォームデータの送信

フォームデータのリセット

フィールドの表示 / 非表示

WWW リンクを開く

ビューアメニューの実行（印刷 / フォームデータのインポート）

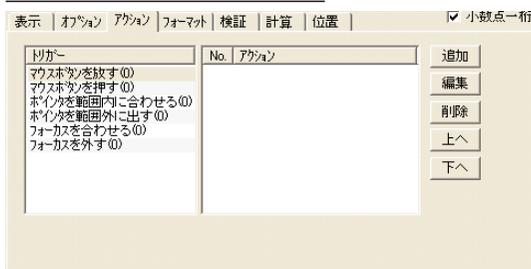
各アクションの詳細については次からの項目をご覧ください。

フィールドにアクションを組み込む

ビューアアプリケーション上でフィールドに対してマウス操作や Tab キーによる移動などの操作が行われた際に動作するアクションを組み込むことができます。

フィールドにアクションを組み込むには、フィールドオブジェクト属性ダイアログ内の [アクション] セクションで設定することができます。

図：フィールドのアクション設定

**[トリガー]**

アクションを設定するトリガーを選択します。

[アクション]

選択したトリガーに登録されているアクションが表示されます。

[追加]

選択したトリガーにアクションを登録します。

登録するアクションの詳細な設定方法は、以降の内容をご覧ください。

フィールドアクションを設定できるトリガーは以下のとおりです。

- マウスボタンを放した時（最も頻繁に使用されます）
- マウスボタンを押した時
- マウスポインタをフィールド領域内に合わせた時
- マウスポインタをフィールド領域外に出した時
- フォーカスを合わせた時
- フォーカスを外した時

※サブミットボタンフィールド、リセットフォームボタンフィールドには既に固定のアクションが実装されているため、フィールドアクションを組み込むことはできません。

[編集]

選択中のアクションの内容を変更します。

[削除]

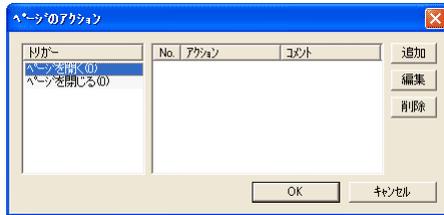
選択中のアクションを削除します。

ページにアクションを組み込む

ビューアアプリケーション内において PDF ファイル内のページの制御が切り替った際に動作するアクションを組み込むことができます。

ページにアクションを組み込むには、Form エディタのメニューから [オプション]-[アクション]-[ページのアクション...] を選択することで表示されるアクション設定ダイアログ上で設定することができます。

図：[ページのアクション]



※アクション設定ダイアログの内容は前述の「フィールドにアクションを組み込む」をご覧ください。

ページアクションを設定できるトリガーは以下のとおりです。

- ページを開く
- ページを閉じる

アクションを組み込む

アクション設定ダイアログから [追加] ボタンを押下するとアクションの選択ダイアログが表示されます。

図：[アクションの選択]



設定したいアクションを選択した後に [OK] を押下すると、各アクションの設定ダイアログが表示されます。各アクションの詳細な設定については以降の項目を参照してください。

JavaScript を組み込む - JavaScript の実行

トリガーによって任意の動作を行わせたい場合には、その動作内容を JavaScript でプログラムし、アクションに組み込むことで実現することができます。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで「JavaScript の実行」を選択し、[OK] を押下することで [JavaScript の編集] ダイアログが表示されます。[JavaScript の編集] ダイアログについては、「5-7. JavaScript を使用する」をご覧ください。

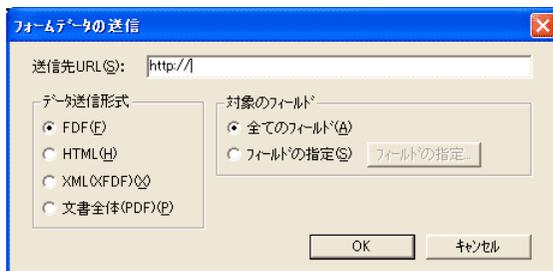
フィールドデータを送信する - フォームデータの送信

トリガーによって指定した URL にフォームデータの送信を行わせることができます。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで「フォームデータの送信」を選択し、[OK] を押下することで [フォームデータの送信] ダイアログが表示されます。

図： [フォームデータの送信]

**[送信先 URL]**

フィールドデータを処理するサーバスクリプトの URL を指定します。

[データ送信形式]

フィールドデータを送信する際のデータ形式を指定します。

データ形式は、

HTML FDF XML (XFDF) PDF

から指定します。各データ形式の詳細については、「5-3. 指定したサーバスクリプトにフィールドデータを送信する」をご覧ください。

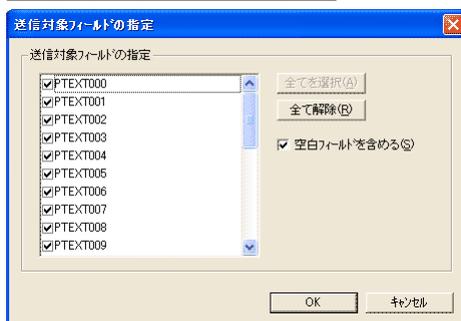
[対象のフィールド]

[送信先 URL] で指定したサーバスクリプトにデータを送信するフィールドを指定します。

全てのフィールドの値を送信する場合には [すべて] を、指定した任意のフィールドの値を送信する場合には [フィールドの指定] を選択します。

[フィールドの指定] を選択した場合は、更に [指定のフィールド] ボタンを押下して [送信対象フィールドの指定] ダイアログ上で送信するフィールドを指定します。

図： [送信対象フィールドの指定]

**[送信対象のフィールドの指定]**

フィールド一覧には送信対象となるフィールド名が表示されています。その左横のチェックボックスではフィールドを送信対象とするかどうかの設定を行います。

[空白フィールドを含む]

この設定が行われていない場合、送信する際にフィールドデータが空のフィールドはサーバスキプトには値が送信されません。設定されている場合には空のデータが送信されます。

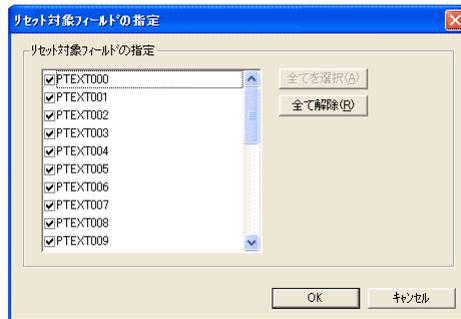
フィールドデータをリセットする - フォームデータのリセット

トリガーによってフィールドに入力したデータをクリアします。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで「フォームデータのリセット」を選択し、[OK]を押下することで[リセット対象フィールドの指定]ダイアログが表示されます。

図：[リセット対象フィールドの指定]



[リセット対象フィールドの指定]

フィールド一覧にはリセット対象となるフィールド名が表示されています。その左横のチェックボックスではフィールドをリセットの対象とするかどうかの設定を行います。

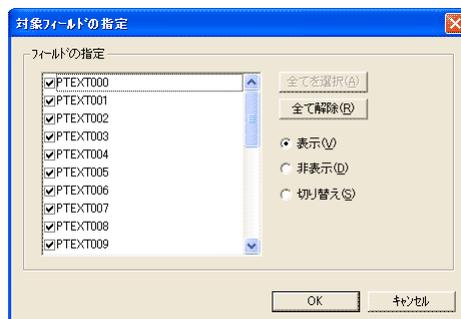
フィールドの表示状態を切り替える - フィールドの表示 / 非表示

トリガーによって任意のフィールドの表示、非表示を切り替えることができます。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで「フィールドの表示 / 非表示」を選択し、[OK]を押下することで[対象フィールドの指定]ダイアログが表示されます。

図：[対象フィールドの指定]



[表示・非表示の指定]

このアクションでは、“表示”、“非表示”、“切り替え”からフィールドの表示、非表示の動作を指定することができます。

表示：非表示状態のフィールドを表示します。

非表示：表示状態のフィールドを非表示にします。

切り替え：非表示状態のフィールドを表示し、表示状態のフィールドを非表示にします。

※ “切り替え” を指定した場合、最大 128 個のフィールドを指定することができます。

[フィールドの指定]

フィールド一覧には表示、非表示切り替えの対象となるフィールド名が表示されています。その左横のチェックボックスではフィールドを表示、非表示切り替えの対象とするかどうかの設定を行います。

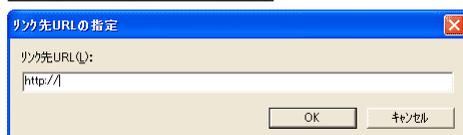
指定した URL に移動する - WWW リンクを開く

トリガーによってインターネットの指定した (http、ftp、mailto プロトコルを使用) URL 先に移動します。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで“WWW リンクを開く”を選択し、[OK] を押下することで [リンク先 URL の指定] ダイアログが表示されます。

図： [リンク先 URL の指定]

**[リンク先 URL]**

移動する先の URL 文字列を指定します。

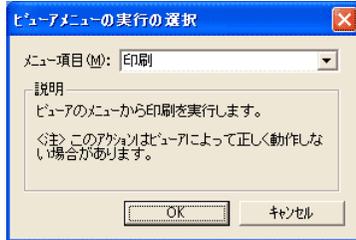
印刷ダイアログを起動する - ビューアメニューの実行

トリガーによってビューアアプリケーションの印刷メニューを実行します。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで“ビューアメニューの実行”を選択し、[OK] を押下することで [ビューアメニューの実行の選択] ダイアログが表示されます。

図： [ビューアメニューの実行の選択] - 印刷



[メニュー項目] から “印刷” を指定します。

※このアクションは、ビューアアプリケーションによっては正常に動作しない場合があります。

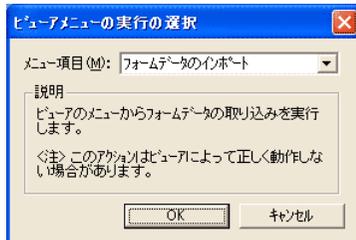
フォームデータをインポートする - ビューアメニューの実行

フォームデータが記述された特定フォーマットファイル (FDF、XFDF など) からフィールドデータを取り込みます。

設定方法

前述の「アクションを組み込む」で述べたようにアクションの選択ダイアログで “ビューアメニューの実行” を選択し、[OK] を押下することで [ビューアメニューの実行の選択] ダイアログが表示されます。

図： ビューアメニューの実行の選択 - フォームデータのインポート



[メニュー項目] から “フォームデータのインポート” を指定します。

※このアクションは、ビューアアプリケーションによって正常に動作しない場合があります。

4. 外部からのデータを反映させる

4-1. フィールドデータを外部データから指定する

テキストフィールドに表示するテキストを外部データから指定する

Datamap エディタ上で対象のテキストフィールドに表示、出力したい文字列の記述位置をマッピングします。

テキストフィールドに対してフォーマットが設定されている場合には、設定に対応したデータが用意されている必要があります。例えば、数値フォーマットが設定されているフィールドに“あいうえお”というデータがマッピングされている場合には、警告レベルのエラーとなりフィールドにデータが出力されません。

同様に検証設定（数値の範囲のみ）が行われている場合においても、設定に対応したデータが用意されている必要があります。

選択状態にするメニュー項目を外部データから指定する

Form エディタで設定したコンボボックスフィールド（もしくはリストボックスフィールド）に登録されているメニュー項目の一覧から選択状態にしたい項目の“書き出し値”（書き出し値の指定がない場合には“項目名”）をマッピングデータとして用意する必要があります。

コンボボックスフィールドのオプションで [編集可能] に設定が行われている場合、マッピングされたデータが登録されたメニュー項目の一覧の書き出し値に一致しない時にはマッピングデータがそのままフィールド上に表示、出力されます。ただし、マッピングされたデータが“空白”の場合、Form エディタ上で設定された初期値が表示されます。

また、コンボボックスフィールドに対してフォーマットが設定されている場合には、設定に対応したデータが用意されている必要があります。例えば、数値フォーマットが設定されているフィールドに“あいうえお”というデータがマッピングされている場合には、警告レベルのエラーとなりフィールドには0が強制的に出力されます。

同様に検証設定（数値の範囲のみ）が行われている場合においても、設定に対応したデータが用意されている必要があります。

コンボボックスフィールド、リストボックスフィールドでは、通常 Form エディタ上で登録するメニュー項目の一覧をマッピングデータから取り込むこともできます。この詳細については、「5-2. メニュー項目を外部データから取り込む」をご覧ください。

チェックマークを入れる項目を外部データから指定する

ラジオボタンフィールド、もしくはチェックボックスフィールドを外部のデータを反映させて選択状態にするには、Form エディタ上のオブジェクトオプションの [書き出し値] で指定されている値に対応したデータが用意されている必要があります。一致しない場合には、出力時に選択状態が反映されません。

フォームデータの送信先 URL を外部データから指定する

サブミットボタンフィールドとボタンフィールドの送信先 URL を外部のデータから指定するには、送信先の URL を示すテキストデータに対してデータマッピングを行います。

Form エディタ上のオブジェクトオプションの [送信先 URL] で送信先が指定されている場合で

も、データマッピングが行われている場合にはマッピングデータが優先的に反映されます。ボタンフィールドにマッピングする場合、複数の「フォームデータを送信」アクションが指定されている場合はその全ての送信先 URL にマッピングデータの URL を使用します。

4-2. フィールドオブジェクトにデータマッピングを行う

Create! FormCollect では、フィールドオブジェクトに対してデータマッピング（データの割り当て）を行うことで、出力時にフィールドにデータを埋め込んだ状態で PDF ファイルを出力することができます。

データマッピングの詳細な説明は、Design マネージャのメニュー [ヘルプ] - [オンラインマニュアル] から「3. 機能リファレンス」-「3.2 エディタの操作」-「Datamap エディタ」をご覧ください。

マッピング方法

フィールドオブジェクトのデータマッピングは他の変数のデータマッピングと同様に Datamap エディタ上で行います。マッピングの方法も他の変数オブジェクト同様に行うことができます。

※リセットフォームボタンフィールド、ボタンフィールド、署名フィールドは、データマッピングを行うことができません。

マッピングデータ

マッピングされるデータとその出力結果はフィールドオブジェクトごと、またはその設定内容によって異なります。以下の項では、フィールドオブジェクトごとのマッピングデータについて説明します。

4-3. データマッピングを行わない場合に反映されるフィールドの値

フィールドオブジェクトは、他の変数と異なりデータマッピングを行わなくても PDF ファイルに出力されます。この場合、出力には各フィールドオブジェクトの初期設定内容が反映されず。

テキストフィールド

Form エディタ上のオブジェクトオプションの [初期値] に指定された文字列が出力されます。

コンボボックスフィールド

リストボックスフィールド

Form エディタ上のオブジェクトオプションの登録されたメニュー項目一覧で初期選択として指定された項目が選択された状態として出力されます。

チェックボックスフィールド

ラジオボタンフィールド

Form エディタ上のオブジェクトオプションの [デフォルトでチェックする] が設定されているオブジェクトが選択された状態で出力されます。設定されていない場合には、未選択状態で出力されます。

サブミットボタンフィールド

Form エディタ上のオブジェクトオプションの [送信先 URL] で指定された URL 文字列がフィールドデータの送信先として設定されます。

5. Tips

5-1. 出力時にフィールドの外観、状態を切り替える - FormSwitch オプション

Create! FormCollect では、出力パターンごとにフィールドの外観等の設定を行うことができ、また出力時に FormSwitch オプション (“-fsN”) を実行コマンドラインに指定することでパターン別に PDF フォームを出力することができます。この機能により同一の資源ファイルを使用して複数パターンの PDF フォームを出力することができます。

出力パターン別にフィールドの外観設定を行う

出力パターンごとのフィールドの外観設定は、各フィールドオブジェクトの属性ダイアログの [表示] セクションで行います。

出力パターン別に設定できる項目は以下のとおりです。各項目の詳細については、「2-2. フィールドの外観、表示状態を設定する」をご覧ください。

境界線の有無	境界線種
境界線幅	境界線色
背景色の有無	背景色
読取専用	表示と印刷

フィールドオブジェクト属性ダイアログの [表示]-[プロパティ]-[出力パターン] 横の [設定] ボタンを押下すると、[出力パターン一覧] ダイアログが表示されます。

図：[出力パターン一覧]



[No.]

出力パターン番号を示します。

[線種]

境界線種を表示します。境界線の描画を行わない設定の場合には“設定なし”と表示されます。

[線幅]

境界線幅を表示します。境界線の描画を行わない設定の場合には“設定なし”と表示されます。

[線色]

境界線色が表示されます。境界線の描画を行わない設定の場合には“x”と表示されます。

[背景]

背景色が表示されます。背景色を設定しない場合には“×”と表示されます。

[読取専用]

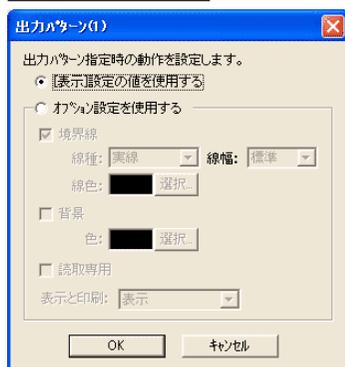
読取専用の設定の有無が表示されます。設定されている場合には“○”、設定されていない場合には“×”が表示されます。

[表示と印刷]

表示と印刷の設定内容が表示されます。

設定したい出力パターン項目を選択して [編集] ボタンを押下（もしくは項目をダブルクリック）すると [出力パターン] ダイアログが表示されます。ここでパターンに対応したフィールドの外観設定を行います。

図： [出力パターン]



初期状態では、[表示] 設定での値を反映するようになっています。パターン別の設定を行うには [パターン設定] を選択し、各項目の設定を行います。各項目の設定方法、内容は前述の項目をご覧ください。

出力パターン別に PDF フォームのプレビューを行う

前項「出力パターン別にフィールドの外観設定を行う」で設定したフィールドの各パターンの出力結果を Form エディタの PDF プレビュー機能で確認することができます。

PDF プレビューを確認するには、Form エディタのメニューから [オプション] - [PDF プレビュー] を選択します。フォーム内にフィールドオブジェクトが定義されている場合には、出力パターンを指定するために [出力パターン指定] ダイアログがプレビュー前に表示されます。

図： [出力パターン指定]



ここで [出力パターン番号の指定] を設定せずに [プレビュー] ボタンを押下すると、出力パターン設定を使用せずに通常の設定を用いた出力プレビューが行われます。[出力パターン番号の指定] を設定し、確認した出力パターン番号を指定して [プレビュー]

ボタンを押下すると、指定した出力パターンの設定が反映されたプレビュー出力を行うことができます。

図：パターン別プレビュー例

通常

FormSwitchオプションサンプル

パターン1

FormSwitchオプションサンプル

パターン2

FormSwitchオプションサンプル

出力パターン別に PDF ファイルの出力を行う

アプリケーションから Create! FormCollect を呼び出して PDF ファイルを生成する際に前項までに設定した出力パターンごとのフィールドの外観設定を反映させるには、実行コマンドラインに FormSwitch オプション (“-fsN”) を指定して Create! FormCollect を呼び出し実行します。

FormSwitch オプション (“-fsN”) の “N” には 1 ~ 10 の出力パターン番号を指定します。指定されない場合 (“-fs” のみ) には、“-fs1” と同じ出力が行われます。

図：通常の実行例

自動車総合保険申込書 < 保険申込書代理店控
兼保険料領収書写し > [送信] [クリア]

郵便番号	160-0023	電話番号	03-3360-6691
(フリガナ) 住所	東京都新宿区西新宿7-5-25		
(フリガナ) 氏名	帳票 太郎		

THE ITM IT 油

図：FormSwitch オプションを指定した実行例

自動車総合保険申込書 < 保険申込書代理店控
兼保険料領収書写し > [送信] [クリア]

郵便番号	160-0023	電話番号	03-3360-6691
(フリガナ) 住所	東京都新宿区西新宿7-5-25		
(フリガナ) 氏名	帳票 太郎		

THE ITM IT 油

5-2. メニュー項目を外部データから取り込む

コンボボックスフィールドやリストボックスフィールドのメニュー項目は通常、Form エディタ上でオブジェクトのオプション設定で登録を行います。Create! FormCollect ではデータマッピングを行うことで出力時に外部のデータからメニュー項目を登録することができます。

この場合、マッピングデータとして定型のデータを用意する必要があります。

定型データ (TXT/CSV/XML 共通)

```
selected_value;item_name1:item_value1;item_name2:item_value2;...
```

説明

selected_value	選択状態を示す項目の書き出し値
item_name	メニュー項目リストに追加する項目名
item_value	項目名に対応する書き出し値

データの先頭には、通常のコンボボックス / リストボックスのマッピングデータである選択状態とする項目の書き出し値を記述します。そのデータのあとにセミコロン（';'）で区切って項目リストデータを追加していきます。1つの項目データは item_name と item_value をコロン（':'）で区切ったセット（※）で構成されます。複数の項目を登録する場合には、セミコロン（';'）で区切って項目のセットを追加していきます。

※追加する項目データは item_name のみでも可能です。この場合、フィールドオブジェクト内では項目名が書き出し値を兼ねることになります。

メニュー項目を外部データから取り込んだ場合には、フィールド作成時に登録したメニュー項目は無効となります。

図：実行例

■コンボボックス（リストボックス）フィールドオプション設定

項目名	書き出し値
<input checked="" type="checkbox"/> りんご	apple
<input type="checkbox"/> オレンジ	orange
<input type="checkbox"/> バナナ	banana

■データマッピングとマッピングデータ

変数名	種別	属性	
COMBO	フィールド	エホ	grape:いちご:strawberry;ぶどう;grape;キウイ:kiwi;

■出力結果

5-3. 指定したサーバスクリプトにフィールドデータを送信する

サブミットフォームアクションを実行することによって、入力、もしくは選択したフィールドのデータを Web サーバに送信することができます。サーバアプリケーション側では送信されたフィールドデータの形式に合わせて、スクリプトなどデータを処理する仕組みを備えている必要があります。

送信されるフィールドデータ

各フィールドの送信されるデータは以下のとおりです。

テキストフィールド

フィールドに入力されたデータが送信されます。[フォーマット] 設定が行われている場合には、フィールド上で表示されている形式ではなく、実際に入力されたデータの形式のまま送信されます。

コンボボックスフィールド / リストボックスフィールド

選択されたメニュー項目に設定されている書き出し値が送信されます。メニュー項目に書き出

し値が設定されていない場合には項目名が送信されます。
コンボボックスフィールドで [フォーマット] 設定が行われている場合には、表示されているデータの形式ではなく、メニュー項目に登録されている書き出し値の形式で送信されます。

チェックボックスフィールド / ラジオボタンフィールド

選択されたフィールドに設定されている書き出し値が送信されます。

データ送信形式とその処理方法の例

フィールドデータを Web サーバに送信する際のデータ形式には以下の種類があります。これらのデータを処理する方法例も合わせて説明します。

PDF 形式

[説明]

PDF (Form Data Format) は Adobe Systems 社の独自フォーマットで、PDF フォームデータを扱うためのファイル形式です。

[処理例]

PDF 形式で送信されたデータを Web サーバ側で処理するには、Adobe Systems 社が提供している FDF Toolkit を利用して解析することができます。FDF Toolkit は PDF フォームにデータを動的に埋め込むための FDF ファイルを構築したり、または FDF ファイルを解析するための API です。C、Perl、Java の API が提供されており、様々なアプリケーションから使用することができます。FDF Toolkit についての詳細は Adobe Systems 社のサイトをご覧ください。

HTML 形式

[説明]

フィールドデータを HTML ファイル形式として送信します。

[処理例]

HTML 形式で送信されたデータを Web サーバ側で処理するためには、HTTP POST メソッドに対応した各スクリプト言語のフィールドデータ取得メソッド（例えば Java Servlet の HttpServletRequest インタフェースで提供されている getParameter (String) 等）を利用することでフィールドデータを取得することができます。

XFDF 形式

[説明]

XFDF は Adobe Systems 社の独自フォーマットで、XML エンコードされた FDF です。XML の構文に則って構成されています。

[処理例]

XFDF 形式で送信されたデータは XML の構文に則ったフォーマットのため、Web サーバ側で処理するためには汎用 XML パーサを使用することでデータの解析を行うことができます。ただし、XFDF のフォーマットを考慮した解析処理を用意しなければなりません。

PDF 形式

[説明]

PDF 形式での送信時には、入力（選択）したデータを含んだ PDF ファイルそのものがサーバ側に送信されます。電子署名を行った場合は、署名情報を含んで送信することができます。ただし、この形式でデータの送信を行うためには送信元（クライアント）で使用しているビューアアプリケーションが Adobe Acrobat である必要があります。その他のビューアアプリケーションが使用される可能性がある場合には、この送信形式は適していません。

[処理例]

PDF 形式で送信されたデータは送信元の PDF ファイルの内容そのもののため、Web サーバ側ではバイナリ保存することでファイルとして送信元と同じ PDF ファイルをサーバ側に保存することができます。

以上は送信後のデータの処理方法の一例です。送信後のデータの扱いに関しては Create!FormCollect のサポートの対象になりませんのでご了承ください。

5-4. ツールチップを表示させる

ビューアアプリケーション上でマウスがフィールドの上にある場合、ツールチップを表示することができます。

設定は、フィールドオブジェクトの属性ダイアログ上で行います。

図： ツールチップの設定



PDFフォーム テキストフィールド	
ワザ外番号: 001	記述: PTEXT
ワザ外名: NAME	
<input checked="" type="checkbox"/> ツールチップ:	名前

[ツールチップ]

ツールチップ機能の使用の有無を設定します。横のテキストボックスにはツールチップとして表示するテキスト文字列を指定します。

図： ツールチップの表示例



帳票 太郎
名前

5-5. Tab キーで移動する順序を指定する

出力した PDF ファイルをビューアアプリケーションで開いた際に、Tab キーを押して移動するフィールドの順序を指定することができます。

この設定は、Form エディタのメニューから [オプション] - [フィールドオブジェクト] - [タブオーダー] を選択して表示される [タブオーダーの設定] ダイアログで行います。

図：[タブオーダーの設定]

**[タブ No.]**

[フィールド名] で示されるフィールドの現在のタブ順序番号が表示されます。

[オブジェクト No.]

[フィールド名] で示されるフィールドのオブジェクト生成番号が表示されます。

[フィールド名]

フォーム上で定義されているフィールド名が全て表示されます。同じ名前が付けられているフィールドが複数フォーム上に存在する場合、ここには複数の同名のフィールド名が表示されます。この場合には [オブジェクト No.] でフィールドを識別するか、フォーム上のフィールドをクリックすることでリスト内の対応するフィールド項目を選択状態にすることができます。

[記述]

[フィールド名] で示されるフィールドを作成した際に属性ダイアログ上の [記述] で指定した文字列が表示されます。

選択したフィールドのタブ順序を先頭に指定する

タブ順序を変更したいフィールドを選択し、[最上位へ] ボタンをクリックします。

選択したフィールドのタブ順序を最後尾に指定する

タブ順序を変更したいフィールドを選択し、[最下位へ] ボタンをクリックします。

選択したフィールドのタブ順序を1つ上げる

タブ順序を変更したいフィールドを選択し、[上へ] ボタンをクリックします。

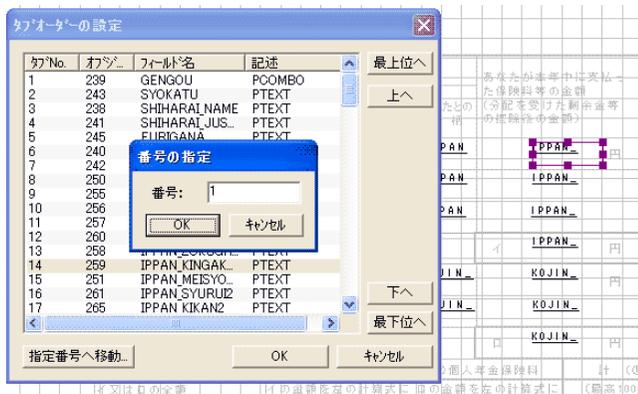
選択したフィールドのタブ順序を1つ下げる

タブ順序を変更したいフィールドを選択し、[下へ] ボタンをクリックします。

選択したフィールドのタブ順序を任意の位置に指定する

タブ順序を変更したいフィールドを選択し、[指定番号へ移動...] ボタンをクリックすると [番号の指定] ダイアログが表示されます。

図：タブ順序 - [番号の指定]



任意のタブ順序番号を入力し、[OK] をクリックしてください。ここで、定義されているフィールド数よりも大きな番号を指定した場合には最後尾にタブ順序が変更されます。

5-6. 計算を行う順序を指定する

いずれかのフィールドで入力、選択が行われた際にテキストフィールド、コンボボックスフィールドで設定されている [計算] 機能は全て動作します。これはフィールドの値が計算結果に影響することを考慮したビューアプリケーション側の挙動です。その際に計算を行う順序によっては期待した結果と異なるものになってしまう場合があります。これに対処するために [計算] が設定されているフィールドを対象とした計算順序の指定を行うことができます。

計算順序の指定手順は、タブオーダーの設定と同様です。Form エディタのメニューから [オプション] - [フィールドオブジェクト] - [計算順序] を選択して表示される [計算順序の設定] ダイアログで行います。

図：[計算順序の設定]



[計算順序]

[フィールド名] で示されるフィールドの現在の計算順序番号が表示されます。

[オブジェクト No.]

[フィールド名] で示されるフィールドのオブジェクト生成番号が表示されます。

[フィールド名]

[計算] 設定が行われているテキストフィールド、コンボボックスフィールド名が表示されず。

[記述]

[フィールド名] で示されるフィールドを作成した際に属性ダイアログ上の [記述] で指定した文字列が表示されます。

選択したフィールドの計算順序を先頭に指定する

計算順序を変更したいフィールドを選択し、[最上位へ] ボタンをクリックします。

選択したフィールドの計算順序を最後尾に指定する

計算順序を変更したいフィールドを選択し、[最下位へ] ボタンをクリックします。

選択したフィールドの計算順序を1つ上げる

計算順序を変更したいフィールドを選択し、[上へ] ボタンをクリックします。

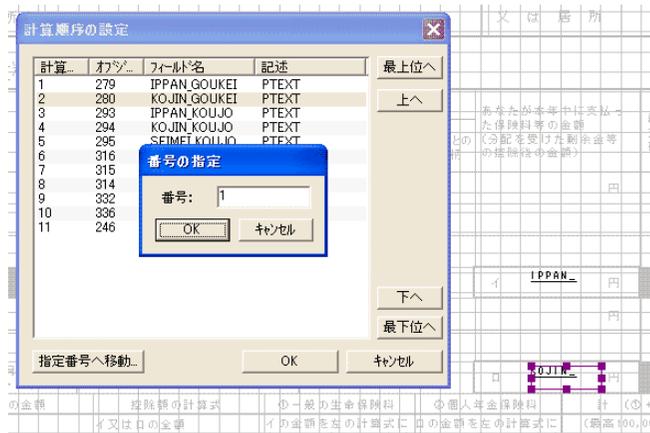
選択したフィールドの計算順序を1つ下げる

計算順序を変更したいフィールドを選択し、[下へ] ボタンをクリックします。

選択したフィールドの計算順序を任意の位置に指定する

計算順序を変更したいフィールドを選択し、[指定番号へ移動...] ボタンをクリックすると [番号の指定] ダイアログが表示されます。

図：計算順序 - [番号の指定]



任意の計算順序番号を入力し、[OK] をクリックしてください。ここで、定義されているフィールド数よりも大きな番号を指定した場合には最後尾に計算順序が変更されます。

5-7. JavaScript を使用する

アクションやフォーマット、検証、計算、項目の選択時の動作、署名後の動作、JavaScript 関数ではカスタム JavaScript を用いることでそれぞれに対応した任意の動作を行わせることができます。カスタム JavaScript の各設定箇所については、以下の各項を参照してください。

アクション	「JavaScript を組み込む - JavaScript の実行」(※1)
フォーマット	「カスタム JavaScript を使用して独自の表示形式を指定する」(※1)
検証	「カスタム JavaScript を使用して独自の検証方法を指定する」(※1)
計算	「カスタム JavaScript を使用して独自の計算方法を指定する」(※1)
項目の選択時の動作	「メニュー項目の選択の変更時に任意の処理を行う」(※1)
署名後の動作	「署名を行った後に任意の動作を行わせる」(※2)
JavaScript 関数	「JavaScript 関数を登録する」(※3)

※1: 「3. フィールドに特殊機能をつける」

※2: 「2. フィールドの作成」

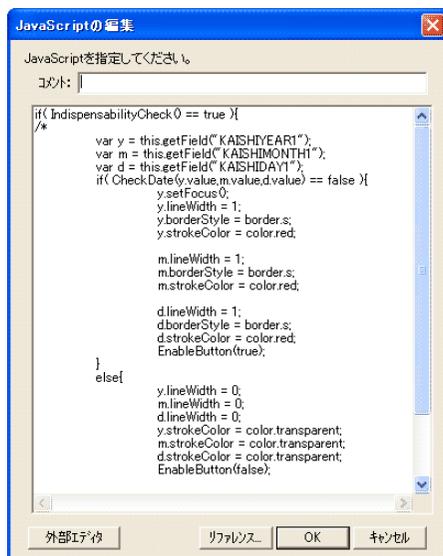
※3: 「5. Tips」

PDF ファイルに組み込むことのできる JavaScript は、Acrobat で動作する Acrobat JavaScript となっています。Acrobat JavaScript については、後述の「Acrobat JavaScript について」をご覧ください。

JavaScript をコーディングする

カスタム JavaScript は、[JavaScript の編集] ダイアログでコーディングを行います。

図: [JavaScript の編集]



[コメント]

この JavaScript アクションに対する任意のコメントを記述します。

[コード]

JavaScript コードを記述します。

このダイアログ上では、30,000 バイトまでのコードを編集することができます。

[外部エディタ]

外部エディタを使用して JavaScript コードを記述します。

外部エディタについては、「5-11. JavaScript を外部エディタを用いて編集する」をご覧ください。

[リファレンス]

既に登録されている JavaScript 関数やフィールドオブジェクト名等をリストしたリファレンスダイアログを表示します。

リファレンスダイアログについては、「5-10. JavaScript リファレンスを使用する」をご覧ください。

[OK] をクリックすることで、JavaScript コードの登録が終了します。

JavaScript をデバッグしたい

Create! FormCollect では設定した JavaScript コードの検証、およびデバッグを行うことはできません。したがって、誤った JavaScript コードを登録した場合もそのまま PDF ファイル内に組み込んで出力を行います。

フォーム設計時に JavaScript の動作を確認するには、[PDF プレビュー] 機能を用いてビューアアプリケーション上で確認します。

[環境]

フォーム設計を行っている環境のビューアアプリケーションに Adobe Acrobat を使用してください。

[手順]

① ページアクションの “ ページを開く ” に対して以下の内容の JavaScript アクションを組み込みます。ページアクションについては、「3-5. ページにアクションを組み込む」をご覧ください。

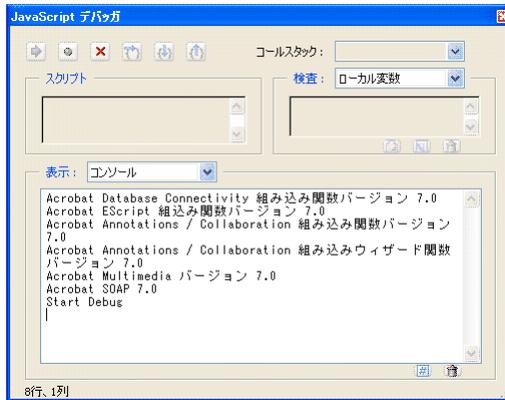
```
console.clear()           // JavaScript コンソールの内容をクリアします。
console.show();           // JavaScript コンソールをビューア上に表示します。
console.println("Start Debug"); // JavaScript コンソール上にメッセージを表示します。
                           // JavaScript 内にデバッグメッセージなどを埋め込む際
                           // に使用します。
```

これらは、PDF ファイルが開かれた際に JavaScript コンソールを表示するコードです。JavaScript コンソールでは、デバッグメッセージを表示することができます。Console オブジェクトの詳細については、後述の「Acrobat JavaScript について」で紹介しているリファレンス資料をご覧ください。

② Form エディタで JavaScript を組み込んだフォームを保存し、[PDF プレビュー] 機能を用いて PDF ファイルを表示する。

③ PDF ファイルが開くと同時に①で設定した JavaScript コンソールが表示されます。

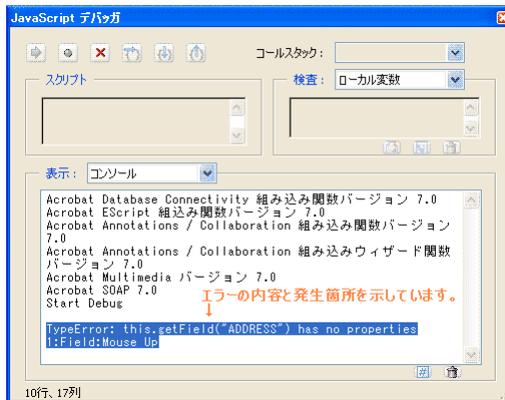
図： Acrobat での JavaScript コンソールの表示



④ 確認したい JavaScript が設定されているトリガーを実行します。

⑤ JavaScript コードに誤りがあるなど実行ができない場合には、③の JavaScript コンソール上にエラー内容が表示されます。

図：エラー内容の表示



Acrobat JavaScript について

JavaScript は、通常ブラウザベースで使用されているコア JavaScript の上に構築されているものですので、Math、String、Date、Array、RegExp といったコア言語機能は全て使用することができます。ただし、ブラウザベースに特化した機能、例えば window や document といったオブジェクト、メソッド、プロパティは Acrobat JavaScript では使用できません。一方、Acrobat JavaScript ではビューアの制御や PDF フォームのオブジェクト（フィールド）へのアクセスが可能となっています。

Acrobat JavaScript の詳細な使用方法は、Adobe Systems 社のサイトから提供されているリファレンス資料を参照してください。

[英語 - Acrobat 7 ベース]

Acrobat JavaScript Scripting Guide

<http://partners.adobe.com/public/developer/en/acrobat/sdk/pdf/javascript/AcroJSGuide.pdf>

Acrobat JavaScript Scripting Reference

<http://partners.adobe.com/public/developer/en/acrobat/sdk/pdf/javascript/AcroJS.pdf>

[日本語 - Acrobat5.05 ベース]

Acrobat JavaScript Object Specification

http://www.adobe.co.jp/support/products/pdfs/acrojs_j.pdf

5-8. JavaScript 関数を登録する

JavaScript コードを関数として登録することができます。JavaScript を関数にまとめ、フォーム内で再利用することでコードの利便性を高めることができるとともに、生成される PDF ファイルの容量が増大することも防ぐことができます。

<< 注意 >>

マルチフォームなど、複数のフォームを1つのジョブで使用する場合は全てのフォーム内で関数名は一意である必要があります。異なるフォームで同一の関数名が定義されている場合には、出力された PDF ファイル内で JavaScript が正常に動作しない可能性がありますのでご注意ください。

JavaScript 関数を管理する

JavaScript 関数を登録（編集、削除）するには、Form エディタのメニューから [オプション] - [アクション] - [JavaScript 関数登録...] を選択します。

図： [JavaScript 関数登録]



[関数名]

登録されている JavaScript 関数名を表示します。

[コメント]

登録されている JavaScript 関数のコメントを表示します。

[追加]

JavaScript 関数を追加します。

[編集]

選択されている JavaScript 関数を編集します。

[削除]

選択されている JavaScript 関数を削除します。

JavaScript 関数を編集する

[JavaScript 関数登録] ダイアログの [追加] ボタン、もしくは [編集] ボタンを押下すると [JavaScript 関数の編集] ダイアログが表示されます。

図 : [JavaScript 関数の編集]

**[関数名]**

この JavaScript 関数名を入力します。新規登録時には関数名が指定されていない場合は JavaScript コードは編集できません。

<< 注意 >>

関数名の先頭は [A-Z, a-z, _] でなければなりません。また先頭以外の文字は [A-Z, a-z, 0-9, _] でなければなりません。

[コメント]

※前述の「5-7. JavaScript を使用する」をご覧ください。

[コード記述部]

※前述の「5-7. JavaScript を使用する」をご覧ください。

[外部エディタ]

※前述の「5-7. JavaScript を使用する」をご覧ください。

[リファレンス]

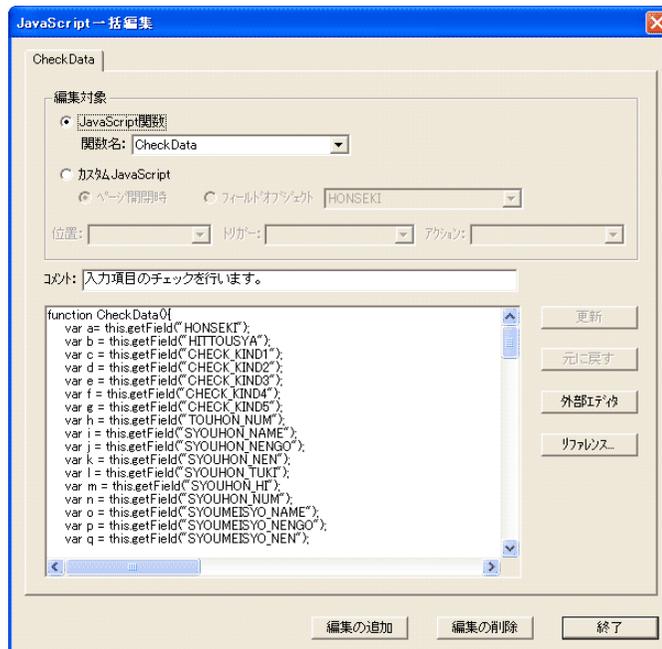
※前述の「5-7. JavaScript を使用する」をご覧ください。

5-9. JavaScript を一括して編集する

Create! FormCollect では、ページやフィールドの様々な箇所に JavaScript を埋め込むことができますが、複雑なフォームになると、どこにどういった JavaScript を埋め込んだか把握しておく必要が生じてきます。このような場合に登録されている全ての JavaScript を参照することができる [JavaScript 一括編集] ダイアログを使用します。

[JavaScript 一括編集] ダイアログを使用するには、Form エディタのメニューから [オプション]-[JavaScript]-[全ての JavaScript の編集...] から表示することができます。

図： [JavaScript 一括編集]



[編集対象]

選択項目を指定して編集する対象のカスタム JavaScript を指定します。

[更新]

カスタム JavaScript の編集内容を更新します。

[元に戻す]

JavaScript コードの内容を編集前の状態に戻します。

編集中に [更新] ボタンが押下された場合には、更新された内容が反映されますのでご注意ください。

[外部エディタ]

外部エディタを使用して JavaScript コードを記述します。

外部エディタについては、「5-11. JavaScript を外部エディタを用いて編集する」をご覧ください。

[リファレンス]

既に登録されている JavaScript 関数やフィールドオブジェクト名等をリストしたリファレンスダイアログを表示します。

リファレンスダイアログについては、「5-10. JavaScript リファレンスダイアログを使用する」をご覧ください。

[ウィンドウを追加]

新しい編集ウィンドウをダイアログに追加します。

[ウィンドウを削除]

アクティブな編集ウィンドウを閉じます。

一括編集ダイアログを使用して JavaScript 関数を編集する

[JavaScript 一括編集] ダイアログの [編集対象] から [JavaScript 関数] ラジオボタンを選択します。[関数名] に現在フォームに登録されている JavaScript 関数が一覧表示されます。その一覧から編集したい JavaScript 関数名を選択すると登録されている関数の内容が編集画面に表示されます。

一括編集ダイアログを使用してカスタム JavaScript を編集する

[JavaScript 一括編集] ダイアログの [編集対象] から [カスタム JavaScript] ラジオボタンを選択します。ここではページアクションやフィールドアクション、フォーマット、検証、計算などに設定されたカスタム JavaScript を編集することができます。

ページに登録された JavaScript を編集する

① [ページ開閉時] ラジオボタンを選択します。

② [トリガー]、[アクション] を指定して、編集対象のカスタム JavaScript を選択します。

フィールドに登録された JavaScript を編集する

① [フィールドオブジェクト] ラジオボタンを選択します。

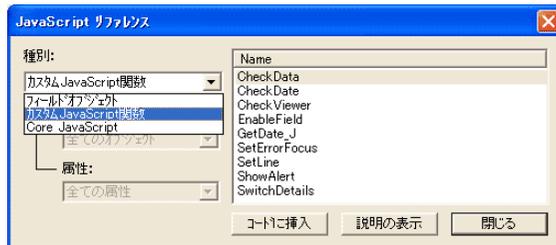
② [位置]、[トリガー]、[アクション] を指定して、編集対象のカスタム JavaScript を選択します。

5-10. JavaScript リファレンスを使用する

JavaScript のコーディングを進めていくとどのような JavaScript 関数が登録されているのか、またはどんなフィールドが定義されているのかなどを参照したい場合が度々生じます。このような状況に対応するために Create! FormCollect ではフォーム上に定義されているフィールドや登録されている関数、更に主だったコア JavaScript のメソッド、プロパティを参照するためのユーティリティダイアログを提供しています。

このダイアログを使用するには、[JavaScript の編集] ダイアログ上にある [リファレンス...] ボタンを押下すると [JavaScript リファレンス] ダイアログが表示されます。

図 : [JavaScript リファレンス]



[種別]

“フィールドオブジェクト”、“JavaScript 関数”、“Core JavaScript” から参照したい種別を選択します。

[オブジェクト]

[種別] で “フィールドオブジェクト”、“Core JavaScript” が選択されている場合のみ有効です。

[種別] で “フィールドオブジェクト” が選択されている場合は、ここで選択されているオブジェクト種別のフィールド名がリストに表示されます。

[種別] で “Core JavaScript” が選択されている場合は、ここで選択されているオブジェクトのプロパティ、またはメソッドがリストに表示されます。

[属性]

[種別] で “Core JavaScript” が選択されている場合のみ有効です

“全ての属性”、“プロパティ”、“メソッド” から選択します。

“全ての属性” が選択されている場合は、登録されている Core JavaScript 項目から [オブジェクト] で示されるオブジェクトのプロパティとメソッドをリストに表示します。

“プロパティ” が選択されている場合は、登録されている Core JavaScript 項目から [オブジェクト] で示されるオブジェクトのプロパティのみを抽出してリストに表示します。

“メソッド” が選択されている場合は、登録されている Core JavaScript 項目から [オブジェクト] で示されるオブジェクトのメソッドのみを抽出してリストに表示します。

[コードに挿入]

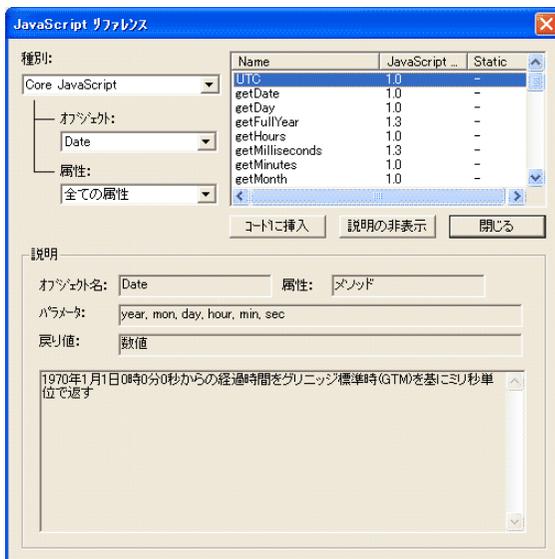
リストから選択した項目を JavaScript コードに挿入します。

選択した項目をリスト上でダブルクリックしても同様の動作を行います。

[説明の表示]

選択した項目の詳細を表示します。

図：説明の表示

**[閉じる]**

この [JavaScript リファレンス] ダイアログを閉じます。

5-11. JavaScript を外部エディタを用いて編集する

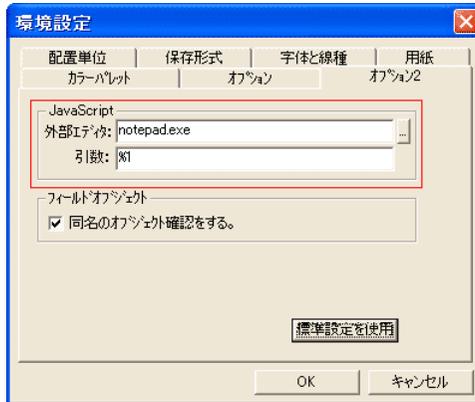
Form エディタ上で JavaScript を作成、編集する際にあらかじめ登録されていた外部エディタを使用することができます。初期状態では、Windows に付属されているメモ帳が外部エディタとして登録されています。

任意の外部エディタを登録する

JavaScript の登録、編集時に使用する外部エディタを登録するには、Form エディタの環境設定で指定することができます。

Form エディタのメニューから [ファイル]-[環境設定] を選択し、[オプション 2] タブの [JavaScript] で任意の外部エディタを指定してください。

図：[環境設定]



[外部エディタ]

使用する外部エディタのパスを指定します。[...] ボタンを押下するとファイル指定ダイアログを利用することができます。

[引数]

外部エディタを起動する際の引数を指定します。

初期値で設定されている "%1" は外部エディタを使用して開くファイルパスを示しています。この前後に使用する外部エディタのオプション引数を指定してください。起動時の引数に関する詳細は、各エディタのヘルプを参照してください。

外部エディタを起動する

JavaScript アクションや JavaScript 関数を登録、編集する際に使用する [JavaScript の編集] ダイアログ、または登録されている JavaScript を一括して編集する [JavaScript 一括編集] ダイアログ上から [外部エディタ] ボタンをクリックすることで登録されている外部エディタを起動させることができます。

編集の場合には、既存 JavaScript コードの内容が外部エディタに反映されます。

起動した外部エディタ上で JavaScript コードを記述した後、ファイルを保存して外部エディタを閉じると呼び出し側のダイアログ上に編集内容が反映されます。

6. 制限事項・変更点

6-1. 制限事項

オブジェクト設定上の制限

■一般

[設定可能数]

変数オブジェクトを含めて最大 999 個

[オブジェクト名]

最大 32 バイト。先頭の文字 [A-Z, 0-9] 以降の文字 [A-Z, 0-9, _, -, #]

[記述]

最大 20 バイト

[ツールチップ]

最大 99 文字

■表示

[フォントサイズ]

2.0 ~ 300.0

■テキストフィールドオプション

[初期値]

最大 9,999 文字

[最大文字数]

最大 32,000 文字

[マス目区切り]

最大 999 文字

■コンボボックス / リストボックスオプション

[登録項目数]

最大 256 項目

[項目名]

最大 9,999 バイト

[書き出し値]

最大 9,999 バイト

[CSV インポート]-[区切り文字 / 制御文字]

半角英数字および半角記号 (ASCII 0x21 ~ 0x3F)

■ラジオボタン / チェックボックス

[書き出し値]

最大 9,999 バイト

■リセットフォームボタン

[表示するラベル]

最大 99 文字

■サブミットボタン

[表示するラベル]

最大 99 文字

[送信先 URL]

最大 9,999 バイト

■検証

[値の検証]-[最小値 / 最大値]

-9,999,999,999,999,999 ~ 9,999,999,999,999,999

■計算

[固定値]

-9,999,999,999,999,999 ~ 9,999,999,999,999,999

JavaScript に関する制限

■共通

[コードサイズ]

最大 30,000 バイト

[コメント]

最大 255 バイト

■ JavaScript 関数

[関数名]

最大 255 バイト

関数名の先頭は [A-Z, a-z, _]。以降の文字は [A-Z, a-z, 0-9, _]

[登録数]

最大 99 個

アクションに関する制限

■登録アクション数

1 トリガにつき 10 アクションまで登録可能

■フィールドの表示 / 非表示

[切り替え] 設定の場合のみ

最大 128 フィールドまで選択可能

数値フォーマットの制限

<< 注意 >>

この制限は Create! FormCollect から出力される際の制限であり、出力後の PDF ファイルをビューアアプリケーション上で入力する場合にはこの制限はありません。

Create! FormCollect では、数値フォーマットが設定されていてデータマッピングが行われているテキストフィールドやコンボボックスフィールドでは、小数部を含む実数部が 15 桁以上のデータを扱うことができません。

6-2. 注意事項

改ページ・マルチフォーム・セット出力に関して

フィールドオブジェクトを含むフォームを使用した出力で複数ページの出力を行う場合には、以下の点にご注意ください。

①改ページ出力

出力される PDF ファイルの構成上、各ページに存在する同名同種のフィールドオブジェクトは同一のオブジェクトとみなされます。これは、複数ページから構成される PDF ファイルの場合、各ページのフィールドオブジェクトの操作が他のページの同名同種のフィールドオブジェクトの内容に反映されることを意味します。

例えば、[NAME] というフィールドが定義されているフォームを用いて出力された 3 ページの PDF ファイル上で 1 ページ目の [NAME] フィールドに入力を行った場合、2、3 ページ目は以下のような内容になります。

図：改ページ時のフィールドの挙動

■ 1 ページ目に「帳票太郎」と入力した場合

2 ページ目

名前：

3 ページ目

名前：

■ 3 ページ目に「帳票花子」と入力した場合

1 ページ目

名前：

2 ページ目

名前：

②マルチフォーム・セット出力

マルチフォーム、セット出力の場合も 1 文書内に同フォームが繰り返し使用されている場合、もしくは異なるフォーム内で同名同種のフィールドオブジェクトが複数定義されている場合には、改ページ出力と同様に PDF ファイル内の同名同種のフィールドは同一のオブジェクトとして扱われます。

ただし、同一のフォームを繰り返し使用せず、かつ複数のフォーム内に同名同種のフィールドが存在しない条件のもとでは、マルチフォーム・セット出力を用いた正常な出力を行うことができます。

また、マルチフォーム・セット出力の場合には、JavaScript 関数名についても考慮する必要があります。フィールド名と同様に使用する全てのフォーム内で関数名を一意にする必要があります。複数のフォームで同一の JavaScript 関数名が定義されている場合には、正常に動作しない可能性があります。

PDF ファイル出力時の JavaScript の動作について

Create! FormCollect で設定した各種のカスタム JavaScript の実行結果は、PDF ファイル出力実行には反映されません。カスタム JavaScript は、ビューアアプリケーションでの PDF ファイル操作時に動作することを前提としていますのでご注意ください。

出力時に結果が反映されないカスタム JavaScript

- アクション（フィールド・ページ）
- フォーマット（書式・キーストローク）
- 検証
- 計算
- 署名後の動作
- 選択の変更
- JavaScript 関数

ビューアアプリケーション上での動作について

Create! FormCollect でのフィールドオブジェクトの設定に基づく要因以外のビューアアプリケーション上での PDF フォームの誤作動、不具合に関しては、弊社のサポート対象とならない場合がございますのでご了承ください。